Questions

- How does Spring Framework Make Unit Testing Easy?
- What is Mockito?
- What is your favorite mocking framework?
- How do you do mock data with Mockito?
- What are the different mocking annotations that you worked with?

```
public class SomeBusinessImpl {
  private DataService dataService;
  //Constructor - public SomeBusinessImpl(DataService dataServ
  int findTheGreatestFromAllData() {
    int[] data = dataService.retrieveAllData();
    int greatest = Integer.MIN_VALUE;

    for (int value : data) {
      if (value > greatest) {
        greatest = value;
      }
    }
    return greatest;
  }
}
```

Basic Mocking

```
@Test
public void testFindTheGreatestFromAllData() {

  DataService dataServiceMock = mock(DataService.class);
  when(dataServiceMock.retrieveAllData())
          .thenReturn(new int[] { 24, 15, 3 });

  SomeBusinessImpl businessImpl =
                  new SomeBusinessImpl(dataServiceMock);

  int result = businessImpl.findTheGreatestFromAllData();

  assertEquals(24, result);
}
```

Basic Mocking

```
@Test
public void testFindTheGreatestFromAllData() {

  DataService dataServiceMock = mock(DataService.class);
  when(dataServiceMock.retrieveAllData())
          .thenReturn(new int[] { 24, 15, 3 });

  SomeBusinessImpl businessImpl =
                  new SomeBusinessImpl(dataServiceMock);

  int result = businessImpl.findTheGreatestFromAllData();

  assertEquals(24, result);
}
```

Using Annotations

```
@RunWith(MockitoJUnitRunner.class)
public class SomeBusinessMockAnnotationsTest {

  @Mock
  DataService dataServiceMock;

  @InjectMocks
  SomeBusinessImpl businessImpl;

  @Test
  public void testFindTheGreatestFromAllData() {
    when(dataServiceMock.retrieveAllData())
            .thenReturn(new int[] { 24, 15, 3 });
    assertEquals(24, businessImpl.findTheGreatestFromAllData()
  }
```

Questions

- What is MockMvc?
- What is @WebMvcTest?
- What is @MockBean?
- How do you write a unit test with MockMVC?
- What is JSONAssert?

Mock MVC Test with Spring Boot

```
public Question retrieveDetailsForQuestion(
      @PathVariable String surveyId,
      @PathVariable String questionId) {
    return
       surveyService.retrieveQuestion(surveyId, questionId);
}
```

Mock MVC Test with Spring Boot

```
@RunWith(SpringRunner.class)
@WebMvcTest(value = SurveyController.class, secure = false)
public class SurveyControllerTest {

  @Autowired
  private MockMvc mockMvc;

  @MockBean
  private SurveyService surveyService;
```

Mock MVC Test - Continued

```
@Test
public void retrieveDetailsForQuestion() throws Exception {
  Question mockQuestion = new Question("Question1",
      "Largest Country in the World", "Russia", Arrays.asList(
          "India", "Russia", "United States", "China"));

  Mockito.when(
      surveyService.retrieveQuestion(Mockito.anyString(), Mock
          .anyString())).thenReturn(mockQuestion);

  RequestBuilder requestBuilder = MockMvcRequestBuilders.get(
      "/surveys/Survey1/questions/Question1").accept(
      MediaType.APPLICATION_JSON);
```

Mock MVC Test - Continued

```
MvcResult result = mockMvc.perform(requestBuilder)
                          .andReturn();

String expected =
  "{id:Question1,description:Largest Country in the World,co

JSONAssert.assertEquals(expected, result.getResponse()
    .getContentAsString(), false);

// Assert
}
```

Questions

- How do you write an integration test with Spring Boot?
- What is @SpringBootTest?
- What is @LocalServerPort?
- What is TestRestTemplate?

Integration Test with Spring Boot

```
@RunWith(SpringRunner.class)
@SpringBootTest(classes = Application.class,
    webEnvironment = SpringBootTest.WebEnvironment.RANDOM_PORT
public class SurveyControllerIT {

  @LocalServerPort
  private int port;

}
```

Integration Test with Spring Boot

```
Test
ublic void testRetrieveSurveyQuestion() {

 String url = "http://localhost:" + port
     + "/surveys/Survey1/questions/Question1";

 HttpHeaders headers = new HttpHeaders();
 headers.setAccept(Arrays.asList(MediaType.APPLICATION_JSON));
 HttpEntity<String> entity =
           new HttpEntity<String>(null, headers);

 TestRestTemplate restTemplate = new TestRestTemplate();
 ResponseEntity<String> response = restTemplate.exchange(url,
     HttpMethod.GET, entity, String.class);
```

Integration Test with Spring Boot

```
    String expected =
    "{id:Question1,description:Largest Country in the World,corr

    JSONAssert.assertEquals(expected, response.getBody(), false)
}
```